INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE

DASC cache: dealing with cache access time and virtual indexing

André Seznec

N° 2082

Octobre 1993

PROGRAMME 1

Architectures parallèles,

bases de données,

réseaux et systèmes distribués

ISSN 0249-6399

DASC cache: dealing with cache access time and virtual indexing

André Seznec *

Programme 1 — Architectures parallèles, bases de données, réseaux et systèmes distribués
Projet Calcpar

Rapport de recherche n ° 2082 — Octobre 1993 — 26 pages

Abstract: In a microprocessor, the cache hit time generally determines the clock frequency. But for the ten last years, a technological trend is the increase of the cache miss penalty in terms of instruction issue delays; then maintaining the cache miss ratio as low as possible is also of particular interest. For a few years, there has been numerous studies focusing on a low cache hit time while maintaining low cache miss ratio. Unfortunately most of the proposed solutions implicitly suppose that the cache is virtually indexed. When using virtually indexed caches, the operating system (or may be some specific hardware) has to manage the consistency of caches and memory.

In this paper, we propose the Direct-mapped Access Set-associative Check cache (DASC) for addressing both difficulties. On a DASC cache, the cache array is direct-mapped then the cache hit time is low, but as the tag array is set-associative,the external miss ratio is the same as the miss ratio of a set-associative cache. When the size of an associativity degree of the tag array is tied to the minimum page size, a *virtually* indexed DASC cache correctly handles all difficulties associated with cache consistency.

Trace driven simulations show that, for cache sizes in the range of 16-32 Kbytes and for current minimum page size in the range 4-8Kbytes, using a DASC cache is a valuable trade-off because it allows fast cache hit time and low cache miss ratio while cache consistency management is performed by hardware.

Key-words: cache hit time, miss ratio, virtually indexed cache, cache consistency

(Résumé : tsvp)

This work was partially supported by CNRS (PRC-ANM)

*seznec@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 38 38 32

L'antémémoire DASC ou le compromis entre l'indexage physique et le temps d'accès

Résumé : Dans ce rapport, nous proposons le modèle de cache DASC (Direct-mapped Access Set-Associative Check).

Dans un cache DASC, le tableau des données est direct-mapped tandis que le tableau des étiquettes (*tags*) est associatif par ensembles. Ceci permet de profiter d'un temps d'accès au cache très bas (tableau de données direct-mapped) tandis que le taux d'échecs sur le cache est maintenu très bas.

Quand l'associavité du tableau de tags est ajustée de telles sortes que la taille associée à un degré d'associativité soit la taille minimum d'une page, l'utilisation d'une structure DASC permet d'indexer le tableau des données avec l'adresse virtuelle et le tableau des tags avec le déplacement dans la page. Ceci permet de profiter à la fois des avantages de l'indexage virtuel (pas de traduction d'adresse avant l'accès au cache) et de l'indexage physique (maintenance automatique de la consistence des données dans le cache).

Mots-clé : cache, temps d'accès, taux d'échec, indexage virtuel, consistence de cache

1 Introduction

In a microprocessor, the cache hit time generally determines the clock frequency [6]. But for the ten last years, a technological trend is the increase of the cache miss penalty in terms of instruction issue delays; then maintaining the cache miss ratio as low as possible is also of particular interest [10]. For a few years, there has been numerous propositions of structures for on-chip caches focusing on reaching a low cache hit time while maintaining low cache miss ratio [22, 3, 10, 2, 5].

Among these structures, the structures based on the use of a direct-mapped caches and possible alternate locations in the cache itself or on the same chip [10, 2, 5] implicitly suppose that the cache is virtually indexed.

When using virtually indexed caches, the operating system (or may be some specific hardware) has to manage the consistency of caches and memory; for enforcing this consistency, a significant part of the potential performance may be wasted in such management on applications sharing pages.

In this paper, we propose the Direct-mapped Access Set-associative Check cache (DASC); on a DASC cache the tag array is set-associative while the cache array is direct-mapped. Then the hit time on the DASC cache is the same as on a direct-mapped cache. Alternate locations to data are provided in the DASC cache, but as the tag array is set-associative, all these locations are checked at the same time.

When the DASC cache has size S, and the minimum page size is P, choosing (S/P) as the degree of associativity of the tag array allows to use the DASC cache with a virtual index, but with all the advantages of a physical indexed cache.

The remainder of the paper is organized as follows. In section 2, we recall some previous propositions of cache structures whose common goal is to reach both low cache hit time -for enabling high clock speed-, and low cache miss ratio -for minimizing performance loss due to miss penalties-. Then, in section 3, we show the drawbacks of virtually indexed caches and point out that among the propositions listed in section 2, only the propositions based on optimistic execution associated with a set-associative cache allow physical indexing.

Then in section 4, we present the DASC cache organization; we particularly show how using a DASC cache structure may allow low cache hit time, low cache miss ratio, virtual indexing **and** an automatic hardware management of cache consistency.

Trace driven simulation results are given in section 6; these results tend to show that for sizes of caches in the 16-32 Kbytes range and for a minimum page size in the 4-8 Kbytes range, the DASC cache organization is a particularly interesting design trade-off. Section 7 summarizes this study.

2 Conjugatind low cache hit time and low cache miss ratio

In this section, we recall some previous propositions of cache structures for allowing low cache hit time and low cache miss ratio.

In this paper, we shall adopt the same definitions as Hill in [6]:

Definition 2.1 *The cache hit time H_c is the delay for getting back a data from the cache on a hit.*

Definition 2.2 *The cache access time A_c is the average delay for getting back a data from the cache.*

A rough modelization of the *cache access time* is:

$$A_c = H_c + miss * Penalty \quad (1)$$

where *miss* is the miss ratio and *Penalty* is the average penalty paid on a miss.

The miss penalty paid when accessing the external memory hierarchy does not depend on the cache structure, but mainly on the design of the memory system. Then efforts in cache design have to be focused on both reducing the cache access time H_c and the cache miss ratio *miss*.

2.1 Direct-mapped caches: low cache hit time

In [6], Hill argued that, in most microprocessor designs, the cache hit time determines the clock of the system: in most microprocessors, the cache hit time is a single cycle.

On a direct-mapped cache, the read of a data may be decomposed in two consecutive steps (figure 1):

1. Read of the word in cache
2. Check the tags against the referenced address

On a n-way set-associative cache, the read of a data is decomposed in three consecutive steps (figure 2):

1. Read of a set of n words in the cache
2. n parallel tag checks against the address of the data
3. selection of the correct word in the set

This extra step explains why the cache hit time is slightly higher on a set-associative cache than on a direct-mapped cache, but for on-chip caches such a difference might be quite unsignificant (2 % were reported by Hill[6]).